2008

Online generation of quasi-optimal spacecraft rendezvous trajectories

Bevilacqua, R.

Acta Astronautica (In Press)
http://hdl.handle.net/10945/38492

ARTICLE IN PRESS

Available online at www.sciencedirect.com

ScienceDirect

PERGAMON

Acta Astronautica ▮▮▮ (▮▮▮▮) ▮▮▮–▮▮▮

ACTA ASTRONAUTICA

www.elsevier.com/locate/actaastro

Online generation of quasi-optimal spacecraft rendezvous trajectories

R. Bevilacqua[a,*], M. Romano[b], O. Yakimenko[c]

[a]*Department of Mechanical and Astronautical Engineering, Naval Postgraduate School, Code AA/RB, 699 Dyer Road, Monterey, CA 93943, USA*

[b]*Department of Mechanical and Astronautical Engineering, Naval Postgraduate School, Code MAE/MR, 700 Dyer Road, Monterey, CA 93943, USA*

[c]*Department of Mechanical and Astronautical Engineering, Naval Postgraduate School, Code MAE/Yk, 700 Dyer Road, Monterey, CA 93943, USA*

Received 22 April 2008; received in revised form 10 July 2008; accepted 8 August 2008

Abstract

A direct method for rapid generation of combined time-propellant near-optimal trajectories of proximity maneuvers of a chaser spacecraft required to dock a target one, with predetermined thrust history along a master direction, is presented. The predetermined thrust history is generated by applying the Pontryagin maximum principle. The new direct method, already implemented and tested on board real aircraft, is based on three concepts: high-order polynomials as reference functions, preset on–off sequence of a master control, and reduction of the optimization problem to the determination of a small set of parameters. Presetting the master control, the remaining controls act as slaves, guarantying the chaser to move along the desired path. Seeking of the optimum strategy is transformed into a nonlinear programming problem, and then numerically solved through an ad hoc algorithm in accelerated time scale. Examples are reported to prove the rapidness of the approach to generate a sub-optimal docking trajectory.
Published by Elsevier Ltd.

Keywords: Online optimization; Direct methods; Spacecraft rendezvous

1. Introduction

Many researches have been performed either for modeling [1–7] or for maneuver optimization [8] of spacecraft rendezvous and docking, but real-time implementation of the optimal control is still a difficult task. In Refs. [9,10] the maneuvers for passing from an initial stable relative motion to a different final stable one are optimized. In these works, there is no possibility of considering generic initial conditions for the relative position and velocity of the chaser vehicle with respect to the target. In Ref. [8] the optimization is performed in two stages: the first part of the trajectory is optimized without any restriction on the chaser vehicle position, the last phase of docking is along a fixed direction.

The importance of this research field can be found in the most recent applications of satellites proximity flight. A very representative example is the ATV developed by the European Space Agency. Current docking

* Corresponding author. Tel.: +1 831 656 2500; fax: +1 831 656 2313.
E-mail addresses: rbevilac@nps.edu (R. Bevilacqua), mromano@nps.edu (M. Romano).

0094-5765/$ - see front matter Published by Elsevier Ltd.
doi:10.1016/j.actaastro.2008.08.001

Nomenclature

Abbreviations and acronyms

ATV	automated transfer vehicle
DMRP	direct method for rapid prototyping
ISS	international space station
LVLH	local vertical local horizontal
NLP	nonlinear programming
OP	optimization parameter
PMP	Pontryagin maximum principle

Symbols

a_{ik}	coefficients of reference polynomials
φ	Euler angle between the velocity vector and the x axis in LVLH frame
g	acceleration due to gravity
J, J_{sc}	performance index, scaled performance index
$k = \{x,y,z\}$	Cartesian coordinates in LVLH frame
m	mass
N	number of nodes
n	order of polynomial
O_{sc}	slaves control constraint violation scaling factor
$P_k(\tau)$	polynomial reference function for the Cartesian coordinates
T_{max}	maximum thrust
t	time
t_T^* (τ_T^*)	thrust-off instant (arc)
t_T^{**} (τ_T^{**})	thrust-on instant (arc)
$t_{scaling}$	time scaling factor
θ	Euler angle between the velocity vector and the xy plane in LVLH frame
u_x, u_y, u_z	control accelerations along the three axis
V	speed
V'	speed as function of the virtual arc
V_{sc}	speed scaling factor
w	cost index weighting coefficient
w_p	penalty weighting coefficient
$w_{overflow}$	slaves control constraint violation penalty weighting coefficient
w_{speed}	speed penalty weighting coefficient
x_i, $i = 1, \ldots, 3$	spacecraft mass center coordinates in the LVLH frame
δ_T	throttle position
Δ, Δ_{sc}	penalty function, scaled penalty function
λ	virtual velocity along the virtual arc
τ	virtual arc
$\Delta\tau$, Δt_j	sampling period
$\boldsymbol{\Xi}$	vector of optimization parameters
bold	representing vectors
bold	representing matrices

Subscripts and superscripts

$(.)_j$	quantity pertaining to the jth time node
$(.)'$, $(.)''$, $(.)'''$	arc derivatives
$\dot{(.)}$, $\ddot{(.)}$, $\dddot{(.)}$	time derivatives
$\bar{(.)}$	relative parameter

related missions, such as the space shuttle-ISS mating or the ATV-ISS supply service, do not include any optimization for the rendezvous trajectories.

The proposed strategy is a direct optimization method already used for the control of aircraft (see Ref. [11]). The basic idea is to parameterize the trajectory in a way that allows for independent choices of path and velocity profiles. The proposed approach reduces the functional problem into an NLP, with a small number of OPs. Although this method gives near-optimal instead of optimal solution, its proven robustness makes it a good candidate for on-board real-time implementation.

Furthermore, the algorithm is fast enough to give on-board autonomy to the spacecraft, as the trajectory optimization may be repeated several times during the maneuvering.

The paper is organized as follows: in Section 2 the dynamical model used in this work is derived. Section 3 formally discusses the optimal control problem. Section 4 is dedicated to the reference functions used for the path parameterization. Based on the optimal control

Please cite this article as: R. Bevilacqua, et al., Online generation of quasi-optimal spacecraft rendezvous trajectories, Acta Astronautica (2008), doi:10.1016/j.actaastro.2008.08.001

synthesized in Section 3, Section 5 introduces the direct method of calculus of variations allowing to obtain prototypes of optimal rendezvous trajectories analytically in real-time. Section 6 deals with inverse dynamics to obtain the remaining states and controls, followed by Section 7 presenting the results of computer simulations.

2. Dynamical model

This section describes the classical linear LVLH referred (see Fig. 1) model [1], used through the rest of the paper for representing the relative motion between two satellites flying in close orbits.

The coordinate axes convention is as follows: x points from the attraction body to the virtual satellite, y as the velocity vector of the satellite, and z completes the frame. The three well-known Hill–Clohessy–Wiltshire differential equations of motion are

$$\begin{cases} \ddot{x} - 2\omega\dot{y} - 3\omega^2 x = u_x \\ \ddot{y} + 2\omega\dot{x} = u_y \\ \ddot{z} + \omega^2 z = u_z \end{cases} \tag{1}$$

with

$$\begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = f_2 - f_1 \tag{2}$$

as the relative acceleration due to non-Keplerian forces acting on the vehicles.

Dealing with linear constant coefficients equations, i.e., a system in the form $\dot{\boldsymbol{X}} = \underline{\boldsymbol{A}}\boldsymbol{X} + \underline{\boldsymbol{B}}\boldsymbol{U}$, being $\boldsymbol{X}$ the state (position and velocity: $x,\ y,\ z,\ V_x,\ V_y,\ V_z$), $\boldsymbol{U}$ the control vector ($u_x,\ u_y,\ u_z$), and

$$\underline{\boldsymbol{A}} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3\omega^2 & 0 & 0 & 0 & 2\omega & 0 \\ 0 & 0 & 0 & -2\omega & 0 & 0 \\ 0 & 0 & -\omega^2 & 0 & 0 & 0 \end{pmatrix}$$

$$\underline{\boldsymbol{B}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3}$$

the linear systems theory [12] can be applied to deduce analytical expressions for the uncontrolled and forced evolution of the state.

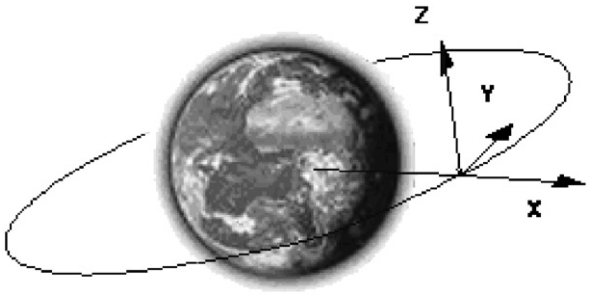

Fig. 1. LVLH reference frame.

At a generic time instant t

$$\boldsymbol{X}(t) = \mathrm{e}^{\underline{\boldsymbol{A}}(t-t_0)}\boldsymbol{X}(t_0) + \int_{t_0}^{t} \mathrm{e}^{\underline{\boldsymbol{A}}(t-\tau)} \cdot \underline{\boldsymbol{B}} \cdot \boldsymbol{U}(\tau)\,\mathrm{d}\tau \tag{4}$$

The forcing term can be analytically solved only in particular cases. The expression for the state transition matrix $\underline{\boldsymbol{\Phi}}(t - t_0) = \mathrm{e}^{\underline{\boldsymbol{A}}(t-t_0)}$, in normalized form, can be found in Ref. [8].

3. Statement of the optimal rendezvous problem

In this section, a general formulation for the mathematical problem is given for the optimization of rendezvous and docking between two satellites.

The physical system is shown in detail in Fig. 2, where a generic maneuver for the chaser agent is represented too.

The target vehicle is assumed to be on a circular orbit, i.e., in the origin of the LVLH frame. It is also assumed that some attitude control devices, such as reaction wheels, are used on the chaser, which is the only controlled agent, to maintain its attitude stable in the LHLV frame. Furthermore, we assume the chaser to be capable of generating the required thrust along the three axis of the LVLH frame. Therefore, as far as we are concerned, the chaser has only three translational degrees of freedom, whose evolution obeys Eq. (4).

In general, it is required to satisfy the following sets of boundary conditions:

$$\begin{aligned} &\boldsymbol{X}(t_0) = \boldsymbol{X}_0 \\ &\boldsymbol{X}(t_f) = \boldsymbol{X}_f = [0\ 0\ 0\ 0\ 0\ 0]^{\mathrm{T}} \end{aligned} \tag{5}$$

The chaser, then, starts from whatever current condition it has and should maneuver itself precisely into the docking position with near-zero velocity.

Possible constraints on states and controls can be included, expressed in general by

$$g_i^{\mathrm{L}} \leqslant g_i(\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{t}) \leqslant g_i^{\mathrm{U}}, \quad i = 1, \ldots, N_{\mathrm{CON}} \tag{6}$$

Please cite this article as: R. Bevilacqua, et al., Online generation of quasi-optimal spacecraft rendezvous trajectories, Acta Astronautica (2008), doi:10.1016/j.actaastro.2008.08.001

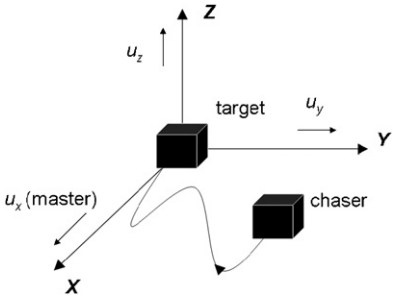

Fig. 2. Chaser and target in LVLH frame.

g_i^{L}, g_i^{U} being, respectively, the lower and upper limits and N_{CON} the number of constraint functions. According to Bolza [13] formulation the optimization problem for system (1) subject to Eqs. (5) and (6) consists in choosing the control input $\boldsymbol{U}$ to minimize

$$J = E(\boldsymbol{X}(t_f), t_f) + \int_{t_0}^{t} f_0(\boldsymbol{X}(t), \boldsymbol{U}(t), t)\,\mathrm{d}t \tag{7}$$

In particular, $f_0 = 1$ for the time minimum problem, $f_0 = u_x^2 + u_y^2 + u_z^2$ for the minimum propellant problem and $f_0 = 1 + w(u_x^2 + u_y^2 + u_z^2 - 1)$ for the combined case here considered, where w is the weighting coefficient. For the case under study, it will be $E(\boldsymbol{X}(t_f), t_f) = 0$.

3.1. Solution for the unconstrained case

The Hamiltonian of the problem with no constraints on states and control variables is

$$\begin{aligned} H = {} & \lambda_x V_x + \lambda_y V_y + \lambda_z V_z + \lambda_{V_x}(2\omega\dot{y} + 3\omega^2 x + u_x) \\ & + \lambda_{V_y}(-2\omega\dot{x} + u_y) + \lambda_{V_z}(-\omega^2 z + u_z) \\ & + w(u_x^2 + u_y^2 + u_z^2 - 1) \end{aligned} \tag{8}$$

Moreover, the set of adjoint differential equations for the co-state vector is [14]

$$\begin{cases} \dot{\lambda}_x = -\dfrac{\delta H}{\delta x} = -3\omega^2\lambda_{V_x} \\ \dot{\lambda}_y = -\dfrac{\delta H}{\delta y} = 0 \\ \dot{\lambda}_z = -\dfrac{\delta H}{\delta z} = \omega^2\lambda_{Vz} \\ \dot{\lambda}_{V_x} = -\dfrac{\delta H}{\delta V_x} = -\lambda_x + 2\omega\lambda_{V_y} \\ \dot{\lambda}_{V_y} = -\dfrac{\delta H}{\delta V_y} = -\lambda_y - 2\omega\lambda_{V_x} \\ \dot{\lambda}_{V_x} = -\dfrac{\delta H}{\delta V_z} = -\lambda_z \end{cases} \tag{9}$$

The adjoint vector shows a linear dynamics very similar to that of HCW equations. More precisely

$$\begin{bmatrix} \dot{\lambda}_x \\ \dot{\lambda}_y \\ \dot{\lambda}_z \\ \dot{\lambda}_{V_x} \\ \dot{\lambda}_{V_y} \\ \dot{\lambda}_{V_z} \end{bmatrix} = -\underline{\boldsymbol{A}}^{\mathrm{T}} \cdot \begin{bmatrix} \lambda_x \\ \lambda_y \\ \lambda_z \\ \lambda_{V_x} \\ \lambda_{V_y} \\ \lambda_{V_z} \end{bmatrix} \tag{10}$$

therefore, likewise Eq. (4) we can deduce

$$\boldsymbol{\Lambda}(t) = \underline{\boldsymbol{\Phi}}_{\lambda}(t - t_0)\boldsymbol{\Lambda}(t_0) \tag{11}$$

Refer to Ref. [8] for the normalized $\underline{\boldsymbol{\Phi}}_{\lambda}$. By deriving Eq. (8) with respect to the controls and by imposing the derivatives to be zero gives the optimal control expression, according to Pontryagin maximum principle (PMP)

$$\begin{bmatrix} u_x^* \\ u_y^* \\ u_z^* \end{bmatrix} = -\frac{1}{2w}\begin{bmatrix} \lambda_{V_x} \\ \lambda_{V_y} \\ \lambda_{V_z} \end{bmatrix} \tag{12}$$

By substituting Eq. (11) (the only part which is needed, i.e., the adjoint velocity) into Eq. (12) and finally into Eq. (4), leads to the optimal unconstrained trajectory

$$\boldsymbol{X}(t) = \underline{\boldsymbol{\Phi}}(t - t_0)\boldsymbol{X}(t_0) + \frac{1}{2w}\underline{\boldsymbol{\Psi}}(t - t_0)\underline{\boldsymbol{\Lambda}}(t_0) \tag{13}$$

where $\underline{\boldsymbol{\Psi}}$ is reported in Ref. [8]. The initial condition on the co-state can be found by matching the boundary conditions at final time

$$\boldsymbol{\Lambda}_0 = \underline{\boldsymbol{\Psi}}^{-1}(t_f - t_0)2w(\boldsymbol{X}(t_f) - \underline{\boldsymbol{\Phi}}(t_f - t_0)\boldsymbol{X}(t_0)) \tag{14}$$

3.2. Solution for the constrained case

In this case, Eq. (6) has to be taken into account by defining an augmented Hamiltonian with respect to Eq. (8), or Lagrangian of the Hamiltonian

$$\overline{H} = H + \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{g}(\boldsymbol{X}, \boldsymbol{U}, t) \tag{15}$$

where $\boldsymbol{\mu}^{\mathrm{T}} = [\mu_1\ \mu_2\ \ldots\ \ldots\ \mu_{N_{\mathrm{CON}}}]$ are the Lagrangian multipliers.

The conditions for a control sequence to be optimal are (1) the derivatives with respect to $\boldsymbol{U}$ be equal to zero and (2) the Lagrangian variables respect the Karush–Kuhn–Tucker (KKT) conditions

$$\mu_i \begin{cases} \leqslant 0, & g_i(\boldsymbol{X}, \boldsymbol{U}, t) = g_i^{\mathrm{L}} \\ = 0, & g_i^{\mathrm{U}} < g_i(\boldsymbol{X}, \boldsymbol{U}, t) < g_i^{\mathrm{L}} \\ \geqslant 0, & g_i(\boldsymbol{X}, \boldsymbol{U}, t) = g_i^{\mathrm{U}} \\ \text{unrestricted}, & g_i^{\mathrm{U}} = g_i^{\mathrm{L}} \end{cases} \tag{16}$$

Please cite this article as: R. Bevilacqua, et al., Online generation of quasi-optimal spacecraft rendezvous trajectories, Acta Astronautica (2008), doi:10.1016/j.actaastro.2008.08.001

The typical existing limitations on engines, forbidden regions on the state space, the presence of obstacles, etc., exclude the possibility of an analytical solution for the constrained case. In other words, there is no closed form solution for the Lagrangian multipliers, as there is for the co-state (Eq. (14)). Depending on the particular problem one numerical method can be preferred among the others.

For the case of limited thrust magnitude (a maximum admissible value $u_{\max}$ exists for the controls) Eq. (12) can be written as

$$u_k^* = \text{sign}(\lambda_{V_k}) \min\left(\frac{|\lambda_{V_k}|}{2w}, u_{\max}\right), \quad k = \{x, y, z\} \quad (17)$$

Eq. (17) highlights the well known bang-unconstrained-bang structure for the fuel optimal case ($w = 1$), and the bang–bang one for the time minimum case ($w = 0$).

4. Introducing the reference trajectory

This section introduces the direct method used for the trajectory optimization in which Eq. (17) is used to describe the behavior of one of the controls. We cannot apply Eq. (17) to the whole control vector, as we do not know the co-state initial conditions for the constrained case.

By starting from Eq. (17) of previous section we will follow the general idea of direct methods for trajectory optimization. The shape of one of the control components (the master) is given along a reference trajectory. The state variables along such trajectory and the remaining controls (slaves) will be calculated by dynamic inversion. The reference trajectory will then be varied to find the "best" solution. This section describes the reference trajectory definition and the following section deals with the predetermined master thrust history.

To be able to separate the path choice from the speed profile choice we use an artificial argument τ (called virtual arc) rather than time t [11]. By doing this we are loosing independency among the control components (in order for the spacecraft to fly along the path, its speed vector should always be tangent to it). However, the a priori introduction of the reference trajectory allows the a priori satisfaction of the boundary conditions (Eq. (5)). It also excludes the possibility of "wild" unpredicted trajectories during the following parameter optimization.

In particular, each of the three chaser's coordinates is represented by a parameterized reference functions of the virtual arc τ, here called $P_x(\tau)$, $P_y(\tau)$, and $P_z(\tau)$, respectively.

We further consider polynomials as reference functions. For instance, to represent the reference trajectory as a fifth-order polynomial, we may write the following:

$$P_k''(\tau) = a_{k2} + a_{k3}\tau + a_{k4}\tau^2 + a_{k5}\tau^3 = \sum_{l=2}^{5} a_{kl}\tau^{l-2}$$
$$k = \{x, y, z\} \quad (18)$$

By integrating Eq. (19) we obtain

$$P_k'(\tau) = \sum_{l=1}^{5} \frac{a_{kl}\tau^{l-1}}{\max(1, l-1)}$$
$$P_k(\tau) = \sum_{l=0}^{5} \frac{a_{kl}\tau^{l}}{\max(1, l(l-1))} \quad (19)$$

The six coefficients a_{kl}, $l = 0, 1, \ldots, 5$ for each $k = \{x, y, z\}$ can be defined by using the following linear matrix equation, together with the boundary conditions:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & \tau_f & \frac{1}{2}\tau_f^2 & \frac{1}{6}\tau_f^3 & \frac{1}{12}\tau_f^4 & \frac{1}{20}\tau_f^5 \\ 0 & 1 & \tau_f & \frac{1}{2}\tau_f^2 & \frac{1}{3}\tau_f^3 & \frac{1}{4}\tau_f^4 \\ 0 & 0 & 1 & \tau_f & \tau_f^2 & \tau_f^3 \end{bmatrix} \begin{bmatrix} a_{k0} \\ a_{k1} \\ a_{k2} \\ a_{i3} \\ a_{k4} \\ a_{k5} \end{bmatrix} = \begin{bmatrix} k_0 \\ k_0' \\ k_0'' \\ k_f \\ k_f' \\ k_f'' \end{bmatrix} \quad (20)$$

To write this matrix equation, the derivatives with respect to time of the coordinates in Eq. (1) were converted to the derivatives with respect to the new argument τ by using the so-called speed factor

$$\lambda = \frac{d\tau}{dt} \quad (21)$$

so that

$$k' = \lambda^{-1}\dot{k}$$
$$k'' = \lambda^{-2}(\ddot{k} - \dot{k}\lambda') \quad (22)$$

Once the coefficients of the reference functions are determined, the reference trajectory will depend only on the parameter τ_f. If additional flexibility is needed in the definition of the trajectory we can increase the order of the reference polynomials and use the higher-order derivatives at both ends as additional varied parameters. For instance, if we use sixth-order polynomials instead of fifth-order polynomials we can use the third derivatives of coordinates with respect to the virtual arc, at both ends of the trajectory, as varied parameters.

Please cite this article as: R. Bevilacqua, et al., Online generation of quasi-optimal spacecraft rendezvous trajectories, Acta Astronautica (2008), doi:10.1016/j.actaastro.2008.08.001

5. Introducing the master control arc history

Let us first convert the system (1) to the new argument. Using the speed factor (21) we can write

$$x' = \lambda^{-1} V_x, \quad V_x' = \lambda^{-1}(2\omega\dot{y} + 3\omega^2 x) + \lambda^{-1} u_x$$
$$y' = \lambda^{-1} V_y, \quad V_y' = \lambda^{-1}(-2\omega\dot{x}) + \lambda^{-1} u_y$$
$$z' = \lambda^{-1} V_z, \quad V_z' = \lambda^{-1}(-\omega^2 z) + \lambda^{-1} u_z \tag{23}$$

Combining the first three equations of Eq. (23) as

$$\sqrt{V_x^2 + V_y^2 + V_z^2} = |\boldsymbol{V}| = \lambda\sqrt{x'^2 + y'^2 + z'^2} \tag{24}$$

specifically addresses the issue of independency of the trajectory and the velocity along it. Having defined the trajectory with respect to the virtual arc τ, i.e., having x', y' and z' defined, still leaves a possibility of varying the magnitude of the speed by varying the speed factor λ. However, the orientation of the speed vector is completely determined by this trajectory (regardless its argument) and can be defined by two Euler angles as

$$tg\,\varphi = \frac{x'}{y'} = \frac{V_x}{V_y}$$
$$tg\,\theta = \frac{x'}{\sqrt{y'^2 + z'^2}} = \frac{V_x}{\sqrt{V_y^2 + V_z^2}} \tag{25}$$

That means that we can no longer vary all of the three controls, u_x, u_y, and u_z, independently. We should define one master control component (in the predominant direction), say u_x, and then define the remaining components so that the direction of the velocity vector is tangent to the trajectory (meaning that equalities (25) hold).

Now, as heuristically suggested by the optimal control theory we assume the master control arc profile to be bang-off-bang as shown in Fig. 3 (with respect to the virtual arc τ rather than time t). Obviously, the remaining controls, u_y and u_z, will not be bang-off-bang anymore. Pulse-width modulation can be used to approximately obtain continuous accelerations in these two channels with on–off actuators [15].

The parameter optimization routine is charted in Fig. 4. Given the boundary conditions (5), we first define the reference polynomials $P_x(\tau)$, $P_y(\tau)$, and $P_z(\tau)$ and compute their coefficients using the boundary conditions (and initial guesses on the third derivatives in case higher than fifth-order polynomials are employed). For the master control, we also establish a bang-off-bang arc profile defined by several switching points. These switching points, τ_i, $i = 1, 2, \ldots, 4$, along with the length of the virtual arc τ_f (and possibly values of the higher-order derivatives of the coordinates at initial and/or final points) form the vector of variable parameters $\boldsymbol{\Xi}$.

Next, we numerically propagate by integrating just one instead of all of the state equations and by applying inverse dynamics for the rest of them. The transition between the virtual arc τ and time t is made by using the speed factor

$$\lambda = \frac{\sqrt{x'^2 + y'^2 + z'^2}}{\sqrt{V_x^2 + V_y^2 + V_z^2}} \tag{26}$$

Then, we estimate the performance index J and compound the aggregated penalty Δ. The existence of this penalty is caused by the fact that the constraints on the two slave controls are necessarily met, since one equation was integrated and two others are related to it via dynamic constraints (25). Consequently the boundary conditions for the final velocity vector components are not satisfied.

Finally, we apply a nonlinear constrained minimization routine to minimize the performance index and keep the penalty within a certain tolerance ε

$$\min_{\boldsymbol{\Xi}} J \mid_{\Delta \leqslant \varepsilon} \tag{27}$$

6. Computation of states, performance index and penalty

This section covers the main algorithmic details of the proposed optimization procedure. We start from dividing the virtual arc τ_f onto $N-1$ equal pieces $\Delta\tau = \tau_f/(N-1)$ so that we have N equidistant nodes $j = 1, \ldots, N$. All states and the master control at the first point $j = 1$ (corresponding to $\tau_1 = \tau_0 = 0$) are defined. Additionally we define $\lambda_1 = 1$.

Then, for each of the subsequent $N-1$ nodes, $j = 2, \ldots, N$ we do the following. We compute the current values of coordinates x, y, and z by using the polynomials $x_j = P_x(\tau_j)$, $y_j = P_y(\tau_j)$, and $z_j = P_z(\tau_j)$, respectively. Next, we integrate the fourth equation of system (23) knowing the master control from the predetermined arc history $u_{x,j-1} = u_x(\tau_{j-1})$

$$V_{x,j} = V_{x,j-1} + \lambda_{j-1}^{-1}(f_1(x_{j-1}, V_{y,j-1}) + u_{x,j-1})\Delta\tau \tag{28}$$

To assure the correct direction of the velocity vector we apply relations (25) to obtain the two remaining

Please cite this article as: R. Bevilacqua, et al., Online generation of quasi-optimal spacecraft rendezvous trajectories, Acta Astronautica (2008), doi:10.1016/j.actaastro.2008.08.001

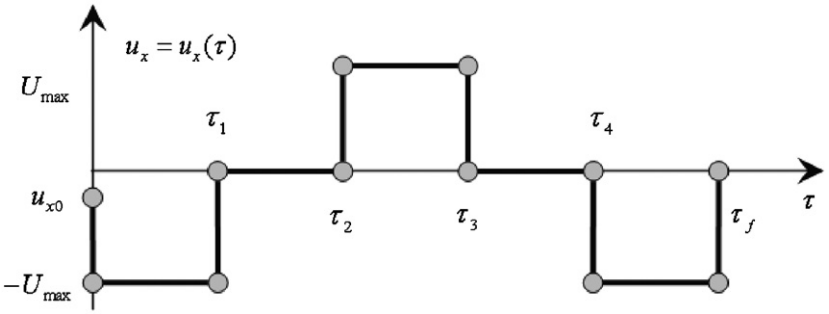

Fig. 3. Control profile for the master control.

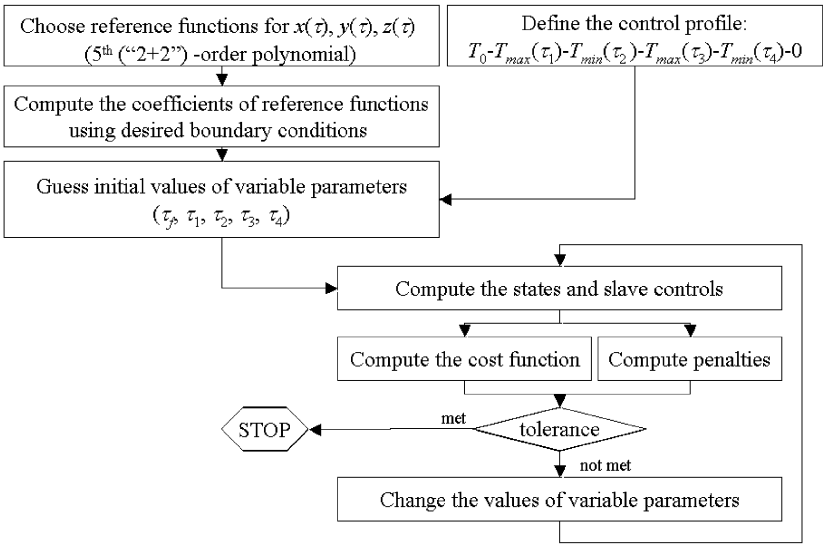

Fig. 4. Flow chart of the parameter optimization.

velocity components

$$V_{y,j} = V_{x,j}\frac{y'_j}{x'_j}$$
$$V_{z,j} = \frac{z'_j\sqrt{V_{x,j}^2 + V_{x,j}^2}}{\sqrt{x_j'^2 + y_j'^2}} \tag{29}$$

Then we calculate the magnitude of the speed

$$|V|_j = \sqrt{V_{x,j}^2 + V_{y,j}^2 + V_{z,j}^2} \tag{30}$$

Now that we know the change in the chaser's position and the magnitude of speed, we may compute the time interval between $(j-1)$th and jth nodes

$$\Delta t_{j-1} = 2\frac{\sqrt{\sum_{i=1}^{3}(k(i)_j - k(i)_{j-1})^2}}{|V|_j + |V|_{j-1}}, \quad k = \{x, y, z\} \tag{31}$$

and the current value of the speed factor

$$\lambda_j = \frac{\Delta\tau}{\Delta t_{j-1}} \tag{32}$$

Current time then equals to

$$t_j = t_{j-1} + \Delta t_{j-1} \quad (t_1 = 0) \tag{33}$$

Finally, by using the last two equations of the system (23), with an inverse dynamics approach, we find the values of the two slave controls that yield the speed components (29)

$$u_{x,j-1} = \frac{V_{y,j} - V_{y,j-1}}{\Delta\tau}\lambda_j + 2\omega V_{x,j-1}$$
$$u_{z,j-1} = \frac{V_{z,j} - V_{z,j-1}}{\Delta\tau}\lambda_j + \omega^2 z_{j-1} \tag{34}$$

Once that the states and controls are computed for all of the nodes, we estimate the performance index

$$J = (1-w)t_N + w\sum_{j=0}^{N-1}(u_{x,j}^2 + u_{y,j}^2 + u_{z,j}^2)\Delta t_j \tag{35}$$

Please cite this article as: R. Bevilacqua, et al., Online generation of quasi-optimal spacecraft rendezvous trajectories, Acta Astronautica (2008), doi:10.1016/j.actaastro.2008.08.001

and the penalty function

$$\Delta = w_p \sum_k (V_{k,N} - \dot{k}_f)^2 + (1 - w_p) \times \sum_k \max_j (0; |u_{k,j}| - U_{max})^2, \quad k=\{x, y, z\} \quad (36)$$

where w_p is the penalty weighting coefficient.

To improve the numerical convergence of the algorithm, the time, the violation of the maximum thrust limits by the slave controls and the discrepancy on final velocity, have been re-scaled. The following modified performance index and penalty was obtained:

$$J_{sc} = t_{sc}(1 - w)t_N + w \sum_{j=0}^{N-1} (u_{x,j}^2 + u_{y,j}^2 + u_{z,j}^2)\Delta t_j$$

$$\Delta_{sc} = w_{speed} V_{sc} \sum_k (V_{k,N} - \dot{k}_f)^2 + w_{overflow} O_{sc} \sum_k \max_j (0; |u_{k,j}| - U_{max})^2 \quad (37)$$

where the coefficients have been adjusted to bring the values to the same order.

7. Simulation results

Three simulation examples are reported to show how the algorithm is capable of generating, in a short time, the command sequence to drive the chaser towards the target in a sub-optimal way.

The boundary conditions are the same for the first two test cases, which assume no out-of-plane displacement in the initial condition. Only the weighting ratio between propellant and time is changed to show how the resulting trajectories differ from each other. In the first case, propellant is more important to be saved than the time required for docking ($w = 0.9$). While in the second test, importance is given to the execution time ($w = 0.1$).

The third simulation has the same parameters set for case 1 ($w = 0.9$), but also an additional 20 m of displacement on the z axis at the initial time.

The numerical values used in the first two simulations are given in Table 1. The Matlab® *fmincon* intended to solve constrained nonlinear optimization problems failed to work in our case. Therefore, the Matlab® *fminsearch* function was used with five varied parameters (i.e., the virtual arc length and the master control switches). And the performance index J_{sc} and penalty function Δ_{sc} (Eq. (37)) blended together.

By online implementation, we mean that the computational time required to produce a feasible quasi-optimal trajectory is a fraction of the duration of this trajectory. In what follows the relative CPU time will be reported for two different machines: an AMD Athlon 2600 MHz processor and a Pentium III 1200 MHz processor running MATLAB/Simulink.

Finally, a cost comparison with a known optimal solution found in literature (Ref. [8]) is presented. The approach of Ref. [8] also considers continuous small thrusters but it does not constraint any of the control channels to behave in an on–off fashion, as in our solution.

It is worth mentioning that in all the simulation runs the fuel cost is computed as commonly done in spacecraft mission analysis, i.e., in the form

$$\Delta V = \sum_{j=0}^{N-1} \sqrt{(u_{x,j}^2 + u_{y,j}^2 + u_{z,j}^2)}\Delta t_j \quad (38)$$

Table 1
Numerical values for the test cases

Parameter	Units	Value
Height above the Earth surface	km	981.46
Initial relative position	m	(−60, −40, 0)
Initial relative velocity	m/s	(0.005, 0, 0)
Required relative final position	m	(0, 0, 0)
Required relative final velocity	m/s	(0.0005, 0, 0)
Initial and final relative accelerations	m/s^2	(0, 0, 0)
Maximum relative acceleration (thrust)	m/s^2	0.0138
Number of points for computation along arc τ	–	200
Time scaling factor	1/s	1/1000
Control constraint violation scaling factor	(m/s^2)$^{-1}$	40
Final discrepancy on velocity scaling factor	(m/s)$^{-1}$	200

Table 2
Significant results of simulation test cases

Parameter	Units	Value
Simulation 1		
Maneuver required time	s	1982
Delta V	m/s	0.898
Maximum control constraint violation	% of max	5.9
Simulation 2		
Maneuver required time	s	3025
Delta V	m/s	0.5
Maximum control constraint violation	% of max	0
Simulation 3		
Maneuver required time	s	3033
Delta V	m/s	0.52
Maximum control constraint violation	% of max	0

Please cite this article as: R. Bevilacqua, et al., Online generation of quasi-optimal spacecraft rendezvous trajectories, Acta Astronautica (2008), doi:10.1016/j.actaastro.2008.08.001

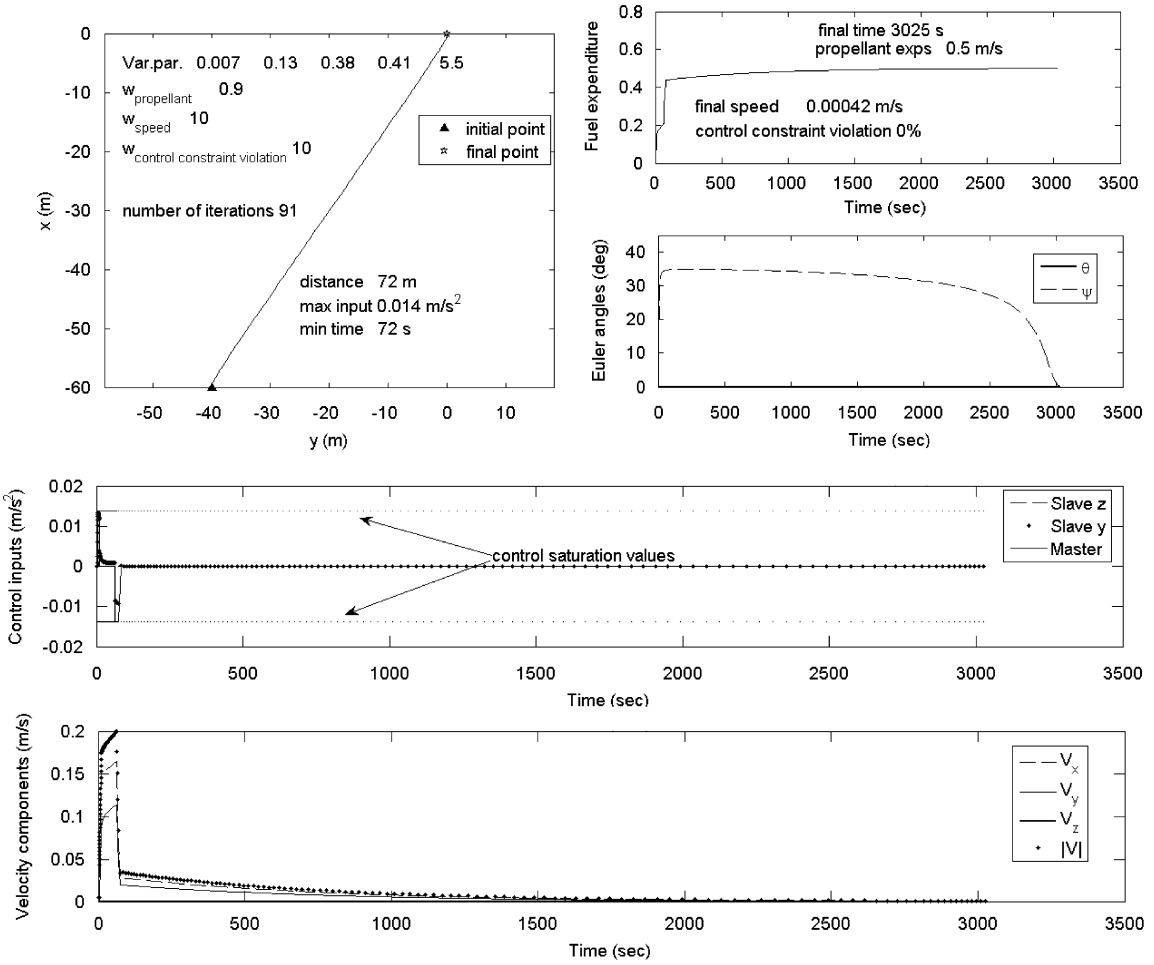

Fig. 5. Results for simulation test case 1.

The main results of the test runs are reported in Table 2.

7.1. Simulation test case 1

The initial guess on virtual arc length and position of the four switching points was $\tau_f = 5.5$, $\tau_1 = 0.007\tau_f$, $\tau_2 = 0.1\tau_f$, $\tau_3 = 0.33\tau_f$, and $\tau_4 = 0.4\tau_f$. The resulting trajectory, with the corresponding optimized values of the OPs controls behavior, velocity history, fuel consumption and other significant parameters is shown in Fig. 5. The controls and velocity are detailed, for the first 300 s of the maneuver, in Figs. 6 and 7.

The number of iterations was 91 (< 100). Note how the final velocity is of the same order of the required one (0.0005 m/s) and the fact that there is no control constraint violation, i.e. the slaves are respecting the imposed bounds. Relative CPU time was in percentage of the maneuver time 2.9% with the faster machine (AMD Athlon 2600 MHz) and 4.3% with the Pentium III, 1200 MHz.

7.2. Simulation test case 2

For this test case, where we still optimize a combination of fuel and time, but giving more importance to the rapidity of the maneuver execution, the initial guess was $\tau_f = 9$, $\tau_1 = 0.007\tau_f$, $\tau_2 = 0.2\tau_f$, $\tau_3 = 0.5\tau_f$, and $\tau_4 = 0.6\tau_f$. The results are shown in Figs. 8–10.

The number of iterations was 84 (< 100). For this maneuver, more demanding than the simulation test case 1, we obtain a control constraint violation of 5.9% for approximately 0.25% of the entire maneuver duration, and the final velocity discrepancy is slightly higher than in the previous case.

Please cite this article as: R. Bevilacqua, et al., Online generation of quasi-optimal spacecraft rendezvous trajectories, Acta Astronautica (2008), doi:10.1016/j.actaastro.2008.08.001

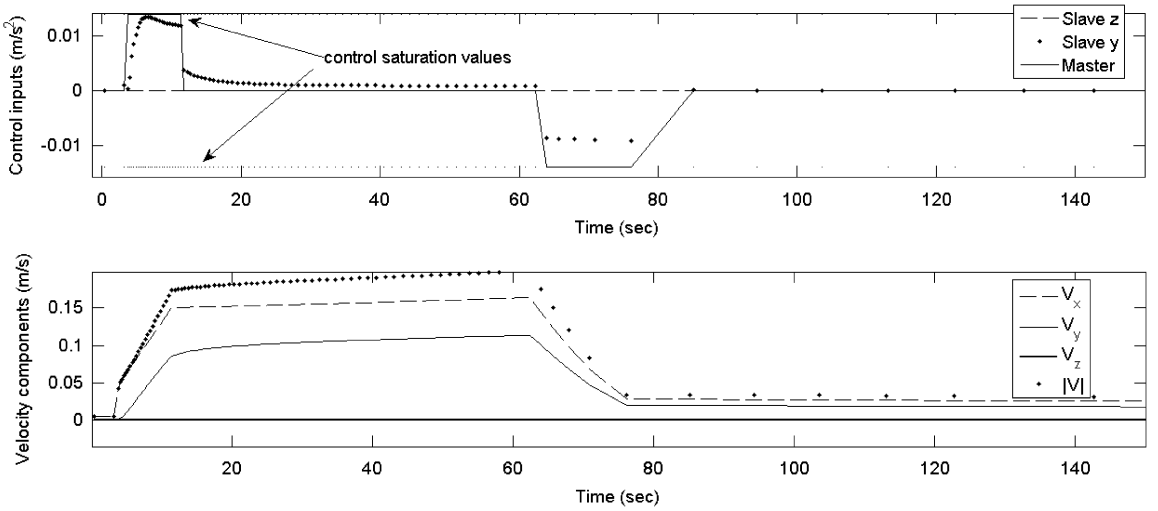

Fig. 6. Zoom of the first 150 s for controls and velocity, test case 1.

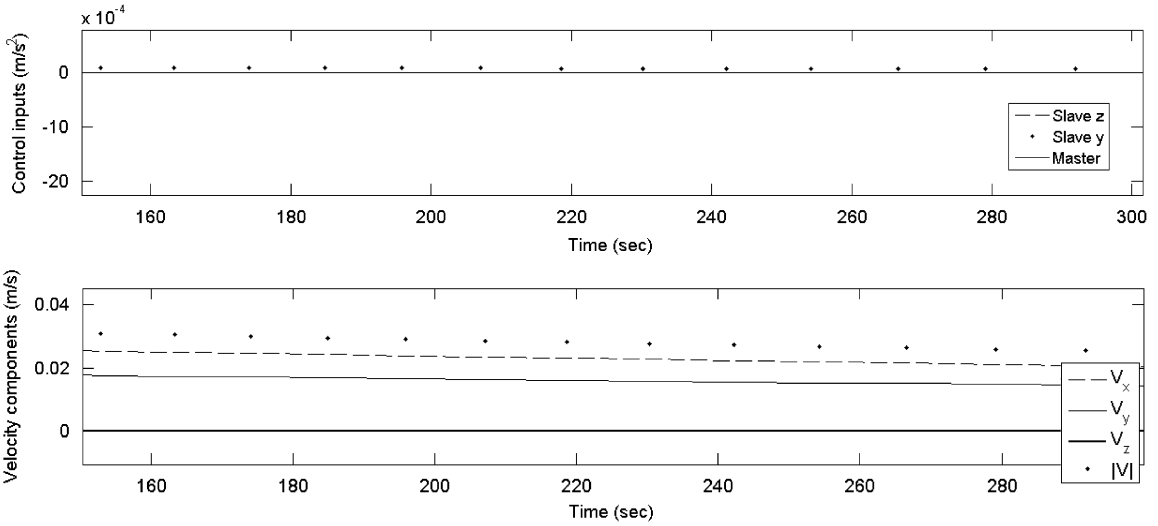

Fig. 7. Zoom from 150 to 300 s for controls and velocity, test case 1.

Having required to minimize time, with a small consideration of propellant expenditure in this case, it results in the possibility of limit violations in the slaves' behavior. The final time is considerably lower than in test case 1, as expected (35%, 1982 s vs. 3025 s). At the same time the propellant expenditure raised from 0.5 to 0.898 units (~80% increase). The relative CPU time was 4.2% on AMD Athlon 2600 MHz and 6% on Pentium III, 1200 MHz.

7.3. Simulation test case 3

An initial displacement of 20 m in the z direction is added with respect to simulation test case 1. The initial guess on virtual arc length and position of the four switching points was $\tau_f = 5.5$, $\tau_1 = 0.007\tau_f$, $\tau_2 = 0.1\tau_f$, $\tau_3 = 0.33\tau_f$, and $\tau_4 = 0.4\tau_f$. The resulting trajectory, with the corresponding optimized values of the OPs controls behavior, velocity history, fuel consumption and other significant parameters is shown in Fig. 11. The 3D trajectory is shown in Fig. 12. The number of iterations was 89 (< 100). Note how the final velocity is of the same order of the required one (0.0005 m/s) and the fact that there is no control constraint violation, i.e., the slaves are respecting the imposed bounds. Relative CPU time is comparable with the that of simulation test case 1.

Please cite this article as: R. Bevilacqua, et al., Online generation of quasi-optimal spacecraft rendezvous trajectories, Acta Astronautica (2008), doi:10.1016/j.actaastro.2008.08.001

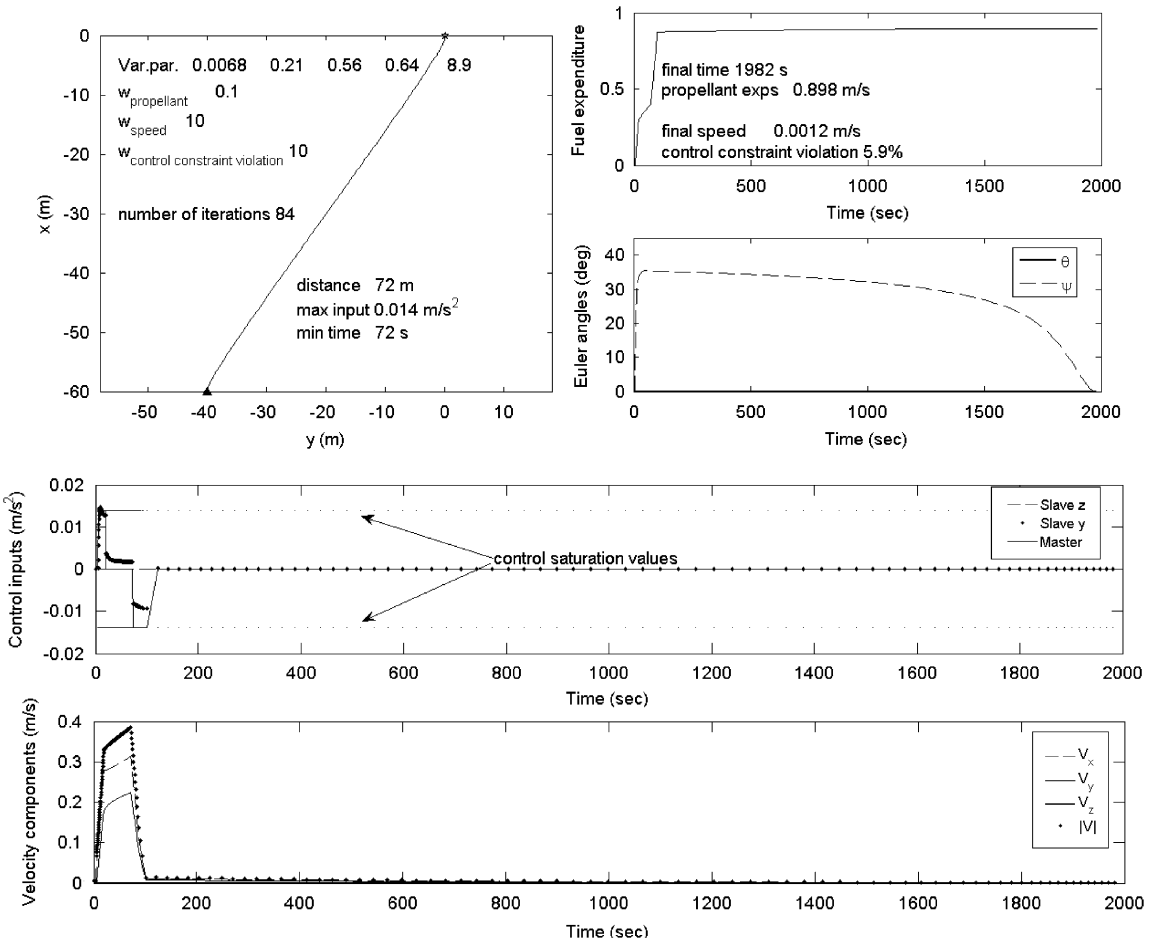

Fig. 8. Results for simulation test case 2.

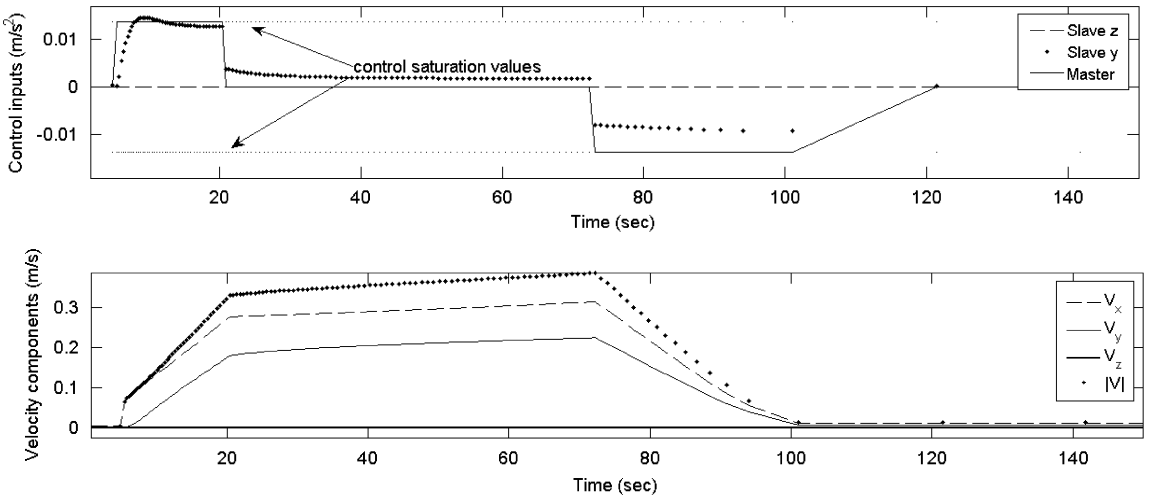

Fig. 9. Zoom of the first 150 s for controls and velocity, test case 2.

Please cite this article as: R. Bevilacqua, et al., Online generation of quasi-optimal spacecraft rendezvous trajectories, Acta Astronautica (2008), doi:10.1016/j.actaastro.2008.08.001

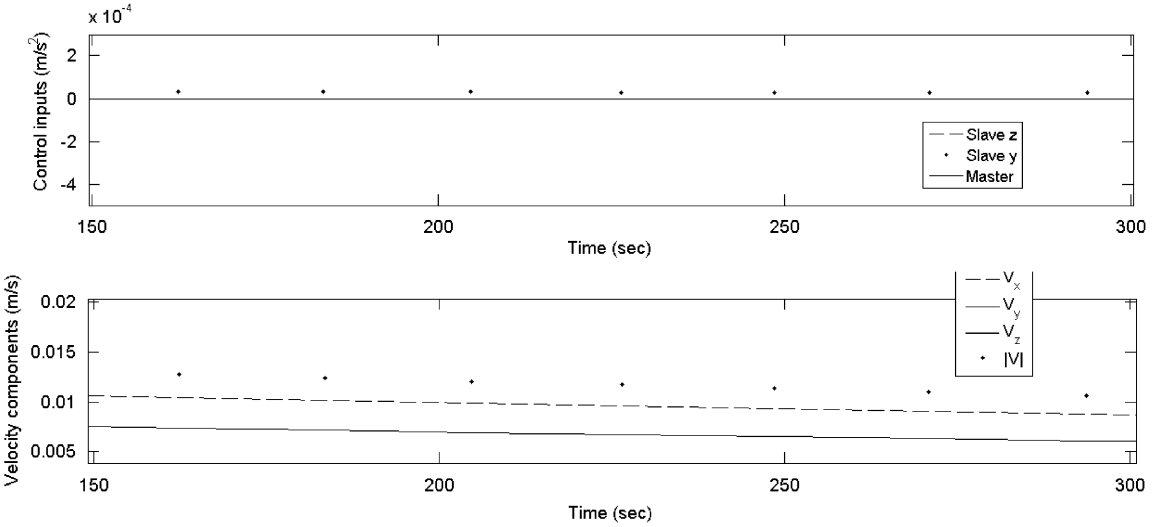

Fig. 10. Zoom from 150 to 300 s for controls and velocity, test case 2.

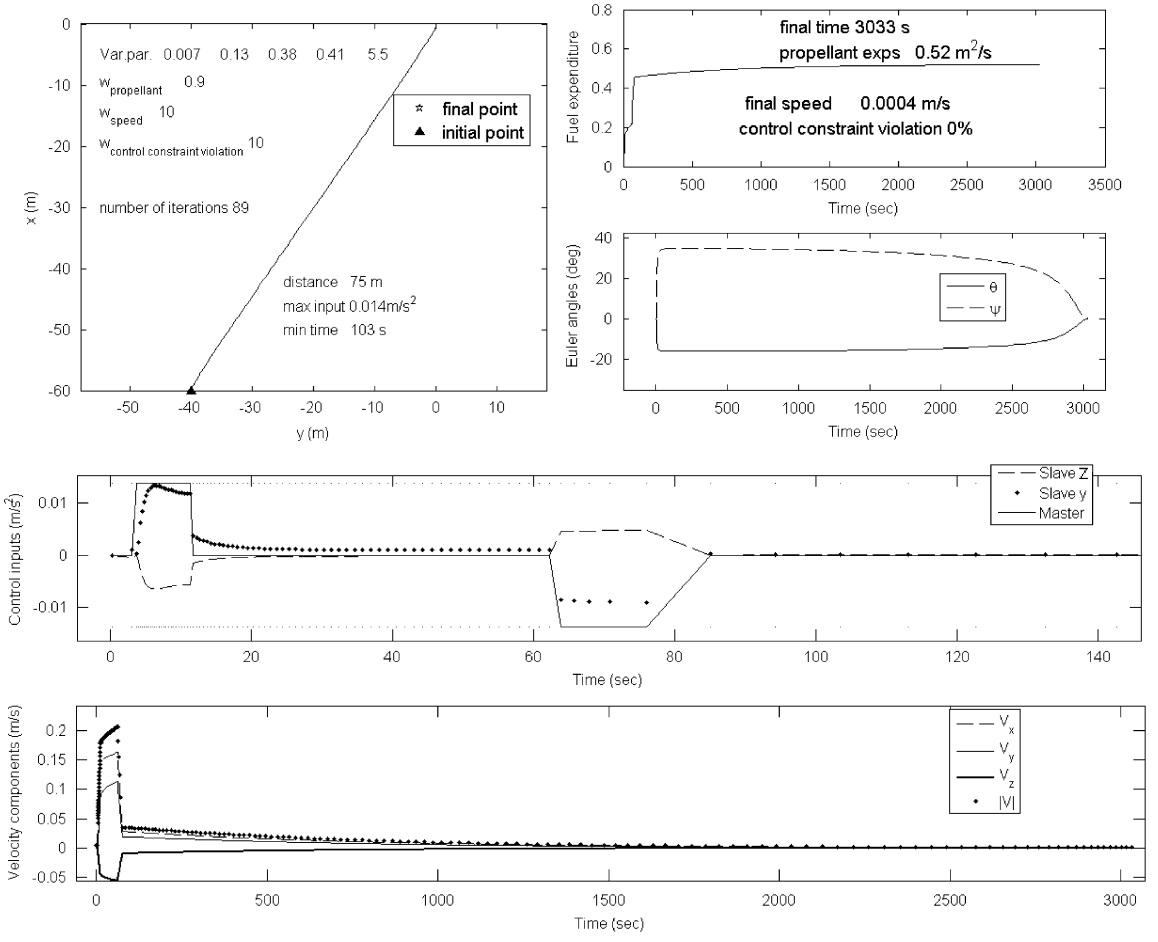

Fig. 11. Results for simulation test case 3.

Please cite this article as: R. Bevilacqua, et al., Online generation of quasi-optimal spacecraft rendezvous trajectories, Acta Astronautica (2008), doi:10.1016/j.actaastro.2008.08.001

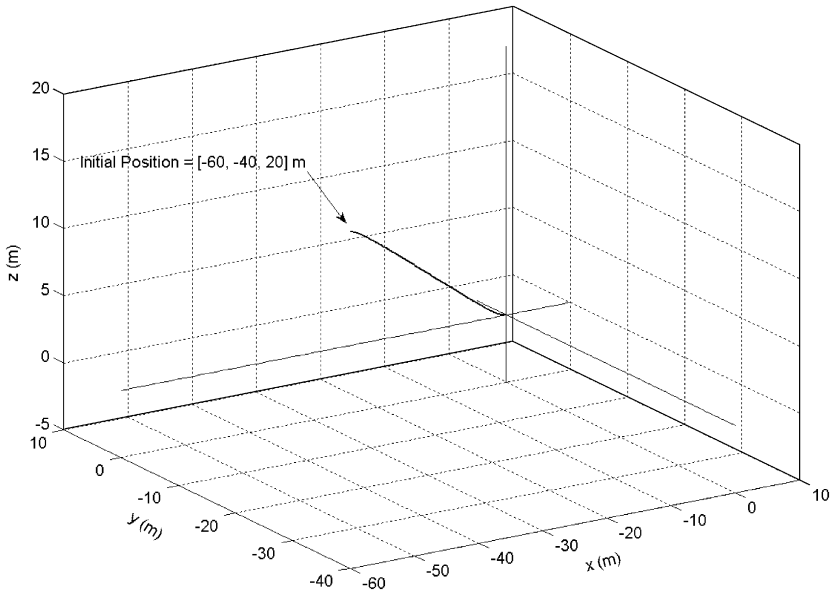

Fig. 12. The 3D trajectory for simulation test case 3.

7.4. Comparison with an optimal solution

The numerical solution of Ref. [8] has been implemented in order to compare the cost obtained with the presented approach. The same boundary conditions, constraint on maximum thrust and maneuver final time of simulation test case 3 are used to run the numerical algorithm described in Ref. [8]. The final ΔV required in the optimal solution is 0.22 m/s, against 0.5 m/s required by our solution. The costs are of the same order of magnitude, and it is worth mentioning once more that the approach of Ref. [8] does not constrain any of the controls to behave as our on–off master; it only binds the maximum value of the thrust components which can vary continuously in all components.

8. Conclusions

The paper introduces an algorithm for the generation of feasible quasi-optimal spacecraft rendezvous trajectories. This algorithm is based on the synthesis of the optimal control via the Pontryagin maximum principle. But rather than numerically solving the two-point boundary-value optimization problem, the algorithm is based on the direct method of calculus of variations and explores the advantages of the inverse dynamics approach, where only one equation is integrated. The proposed direct method has a set of key advantages. First of all it guarantees the boundary conditions to be satisfied (by construction for position, numerically for velocity), no "wild" trajectories arise during optimization, an analytical (parametrical) representation of the reference trajectory is possible, allowing, for instance, to easily account for the presence of obstacles, finally, a small number of OPs have to be considered, requiring only a few iterations (< 100) to generate a solution. A low relative CPU time with respect to the maneuver time is needed, making possible to employ this approach on-board a spacecraft. A comparison with an optimal solution from available literature is also presented.

Further development of the present study will include hardware-in-the-loop tests on the experimental setup available at the Spacecraft Robotics Laboratory of the Naval Postgraduate School [16].

References

[1] W.H. Clohessy, R.S. Wiltshire, Terminal guidance system for satellite rendezvous, Journal of the Aerospace Sciences 27 (9) (1960) 653–658.

[2] J. Tschauner, Elliptic orbit rendezvous, AIAA Journal 5 (6) (1967) 1110–1113.

[3] G. Inalhan, M. Tillerson, J.P. How, Relative dynamics and control of spacecraft formations in eccentric orbits, Journal of Guidance, Control and Dynamics (1) (2002).

[4] D. Izzo, Formation flying linear modeling, in: Proceedings of the Fifth International Conference on Dynamics and Control of Systems and Structures in Space, Cambridge, UK, 14–18 July 2002, pp. 283–289.

[5] D. Izzo, M. Sabatini, C. Valente, A new linear model describing formation flying dynamics under J2 effects, in: AIDAA, XVII Congresso Nazionale, Roma, 15–19 September 2003.

Please cite this article as: R. Bevilacqua, et al., Online generation of quasi-optimal spacecraft rendezvous trajectories, Acta Astronautica (2008), doi:10.1016/j.actaastro.2008.08.001

[6] H. Schaub, K.T. Alfriend, J2 invariant relative orbits for spacecraft formations, in: Goddard Flight Mechanics Symposium, 18–20 May 1999, paper no. 11.
[7] S.R. Vadali, An analytical solution for relative motion of satellites, in: Proceedings of the Fifth International Conference on Dynamics and Control of Structures and Systems in Space, 2002.
[8] M. Guelman, M. Aleshin, Optimal bounded low-thrust rendezvous with fixed terminal-approach direction, Journal of Guidance, Control and Dynamics 24 (2) (2001).
[9] M.E. Campbell, Planning algorithm for multiple satellite clusters, AIAA Journal of Guidance, Control and Dynamics 26 (5) (2003).
[10] S.R. Vadali, S.S. Vaddi, K.T. Alfriend, An intelligent control concept for formation flying satellite constellations, International Journal of Nonlinear and Robust Control, Dedicated to Formation Flying 12 (2002) 97–115.
[11] O.A. Yakimenko, Direct method for rapid prototyping of near-optimal aircraft trajectories, Journal of Guidance, Control, and Dynamics 23 (5) (2000) 865–875.
[12] A. Isidori, Sistemi di Controllo, Siderea, Seconda Edizione, Volume Primo, 1992.
[13] O. Bolza, Lectures on the Calculus of Variations, University of Chicago Press, Chicago, 1904.
[14] L.S. Pontryagin, V.G. Boltjanskiy, R.V. Gamkrelidze, E.F. Mishenko, The Mathematical Theory of Optimal Processes, Wiley-Interscience, New York, 1969.
[15] R. Bevilacqua, M. Romano, Fuel optimal spacecraft rendezvous with hybrid on–off continuous and impulsive thrust, AIAA Journal of Guidance, Control and Dynamics 30 (4) (2007) 1175–1178.
[16] M. Romano, D.A. Friedman, T.J. Shay, Laboratory experimentation of autonomous spacecraft approach and docking to a collaborative target, AIAA Journal of Spacecraft and Rockets 44 (1) (2007) 164–173.

Please cite this article as: R. Bevilacqua, et al., Online generation of quasi-optimal spacecraft rendezvous trajectories, Acta Astronautica (2008), doi:10.1016/j.actaastro.2008.08.001